Service-Oriented Visualization of Virtual 3D City Models

Authors: Jan Klimke, Jürgen Döllner

Computer Graphics Systems Division

Hasso-Plattner-Institut, University of Potsdam, Germany

http://www.hpi3d.de

jan.klimke@hpi.uni-potsdam.de, juergen.doellner@hpi.uni-potsdam.de

Virtual 3D City Models

Virtual 3D city models [1] provide means to communicate complex 2D and 3D geographical and geospatial data in an effective way (Fig. 2). Visualization systems that are based on 3D geovirtual environments (3D GeoVEs) serve as enabling technology in GIS and for manifold fields of application, e.g., as platform for presentation and communication of geodata (e.g., city marketing, urban planning, public participation), as visualization used in navigation, or for 3D computational analysis [3] such as solar potential analysis of roof surfaces (Fig. 1) [4] or urban noise pollution. Due to the massive amounts of data to be processed, stored, and rendered (e.g., ranging from several GB to TB), common applications for 3D geovisualization are running on specialized, high performance hardware take advantage of large main memory, extensive disk capacity and high-performance 3D graphics hardware. As classical kind of desktop application, the software architecture is characterized by its monolithic structure and, consequently, design, implementation, and deployment of 3D applications and systems is faced by a high degree of complexity regarding implementation, portability, robustness, and deployment issues.

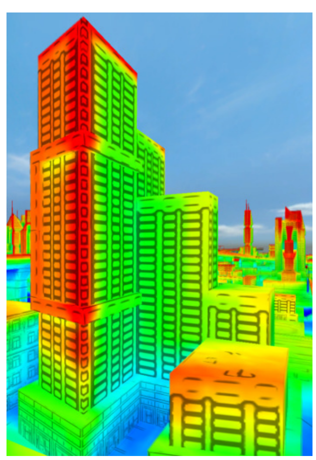

Figure 1 Visualization of a solar potential analysis based on a 3D city model

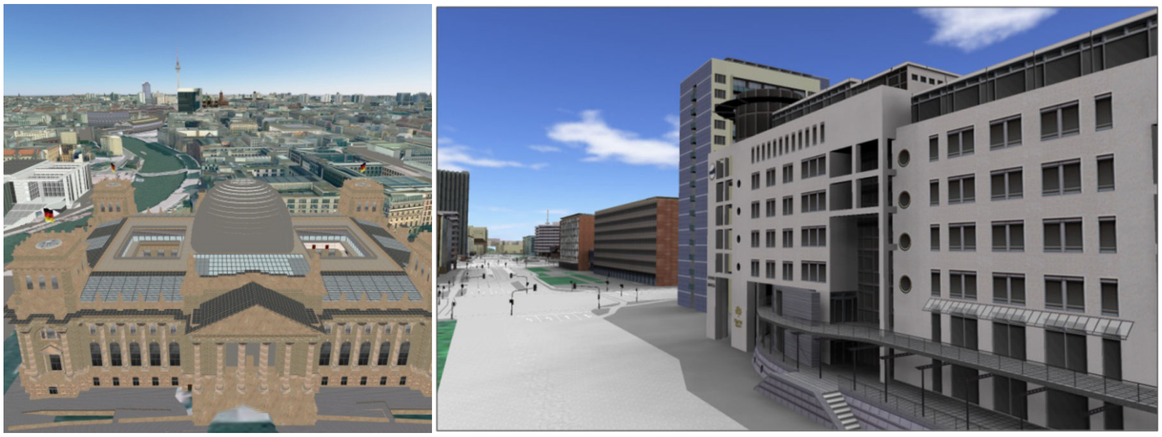

Figure 2 Examples for 3D city model visualization.

Service-Based Visualization of Virtual 3D City Models

The service-based visualization represents a key prerequisite for the future use of virtual 3D city models in applications and systems. The classical approach based on a monolithic application architecture and resulting "fat applications" is not suited for this purpose due to several reasons:

- Data transmission between servers and clients that is limited by network bandwidth
- Massive 3D data sets that can hardly be efficiently stored and managed on clients
- 3D rendering techniques that are restricted to 3D graphics capabilities of clients
- Advanced 3D rendering techniques that can only be partially implemented on clients
- Energy consumption that is drastically increased by 3D rendering computations

A general solution to these challenges can be designed following a service-oriented approach for visualizing virtual 3D city models. In general, service-oriented architectures for 3D geovisualization foster a separation of concerns into management and rendering of virtual 3D city models and their interactive provision by client applications. While services for geodata access (Web Feature Service, WFS), processing (Web Processing Service, WPS) and 2D portrayal (Web Map Service, WMS) are standardized and are successfully used in the context of spatial data infrastructures, service-based management and rendering of 3D models as well as corresponding 3D client applications have not been matured to a similar degree. We refer to these approaches as *3D portrayal services* because the portrayal in the sense of visualization represents their main functionality.

Two types of 3D portrayal services, currently discussed in the OGC context, can be distinguished:

- *Web 3D Service* (W3DS) [7]: It handles geodata access and mapping to renderable computer graphics primitives (e.g., textured 3D geometry represented by scene graphs) and their delivery to client applications.

- *Web View Service* (WVS) [5, 6]: It encapsulates the image generation process (i.e., it implements the geovisualization pipeline) of 3D models, delivering rendered image representations ("portrayals") to client applications.

In the W3DS case, client applications implement and perform 3D rendering, i.e., clients have to process geometry and texture data of 3D city models. Consequently, clients rely on and are limited by their own memory and graphics resources.

WVS Servers

In the WVS case, the server implementation encapsulates the whole 3D rendering process, which has to cope with geometry and texture data of virtual 3D city models; the implementation and deployment can be optimized for a given (controlled) hardware and software configuration. The WVS approach decouples the clients from the complexity of 3D city models – the clients have to handle the image sequences but are not affected by the complexity of the underlying 3D model. Furthermore, the WVS approach allows us to use and optimize advanced 3D rendering techniques in contrast to implementations that have to be compatible to a potentially large hardware diversity of the client devices. The client application, in general, requires only moderate 3D features to reconstruct or display the virtual environment encoded in the image sequences or panorama sequences. This way, the WVS approach facilitates and simplifies the implementation of 3D visualization systems.

WVS Clients

Basic 3D client applications (e.g., a browser-based application to request a single view) retrieve a single image containing a perspective view of the virtual 3D city model from a WVS and provide user interface elements for stepwise 3D navigation, requesting new images for the changed camera parameters. Compared to 3D clients that directly manage and interactively render 3D models (e.g., retrieved from a W3DS) 3D clients based on perspective views can only provide a limited interactive 3D experience to the user. Though, we are generally faced with a trade-off between interactivity (based on client-side 3D rendering) and model complexity (costs for streaming 3D model data from the server to the client) in the scope of service-oriented visualization.

In our approach, we solve that trade-off to a certain degree by generating images of a six-sided box, the *cube map*, transferring these six images to the client, where a specialized 3D rendering application allows users to interactively visualize and explore the resulting virtual 3D panorama. For it, the 3D rendering process, running on the server, creates cube maps of virtual 3D city models. For each image, we do not only create the RGB color layer but also a depth layer and an object-identifier layer. In computer graphics, such multi-layered image is referred to as *G-Buffer*.

The 3D client application uses the cube map to simulate a discretized version of the virtual 3D city model; the user can interactively explore the virtual 3D panorama. If the users zooms in or navigates though the 3D space, the client requests a new version of the cube map. For the 3D client application, the virtual 3D panorama generally represents a lightweight 3D model.

The key advantages of this approach include:

- The 3D client application has not to cope with the original model's complexity because it receives a stream of cube maps with a fixed resolution.
- The 3D server process can implement advanced 3D rendering techniques and use specialized 3D graphics hardware but has not to consider the graphics hardware profiles of the clients.
- Client and server only interchange a sequence of cube maps, encoded by standard image formats (or video formats), and clients can compensate delays in the delivery of cube maps using previous, possible non-optimal cube maps as an intermediate approximation.

Our implementation [2] of such cube-map based client-server 3D rendering approach fetches different information layers (color, camera distance, and object id) as images from a WVS and incorporates two different techniques to build a partial representation of the complete scene using this image data:

- A textured cube to enable a free look around (change of viewing direction);
- A 3D triangle mesh built out of camera distance image layers as intermediate representation for translations of the virtual camera.

3D meshes approximate the server-side 3D model for one specific viewpoint (Fig. 3). They are used to maintain user orientation in case a user moves the virtual camera. After a camera motion stopped, a new cube map representation is request, providing an optimal view on the scene for the new camera position and orientation. Such reduced, lightweight representations of 3D city models on client side allow the interactive client application also to run on mobile devices, such as smart phones and tablets (Fig 4).

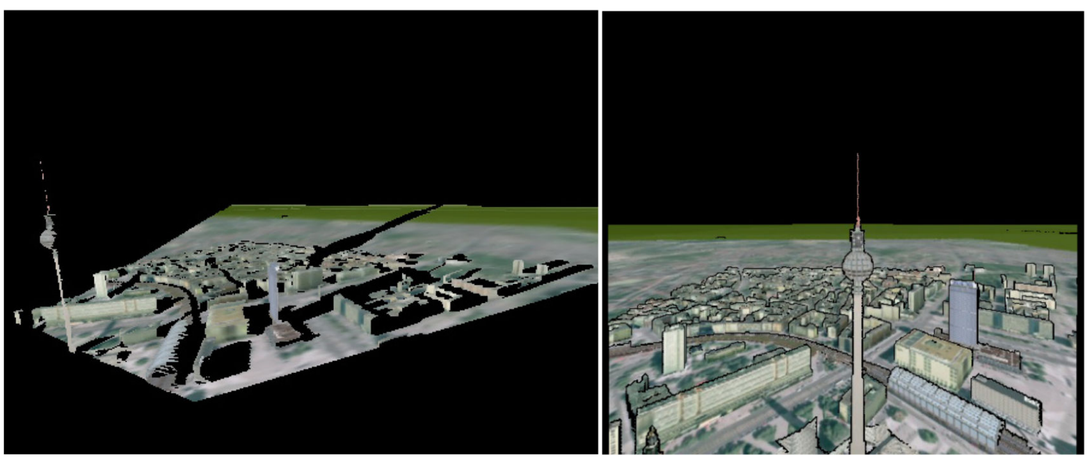

Figure 3 Side view (left) and view from the requested camera position (right) of a mesh reconstruction for one side of a cube map. Depth and color information is used to generate a textured 3D mesh. Areas where no data for reconstruction is available from the retrieved image layers are colored black.

Figure 4. Mobile interactive 3D client application using a WVS to visualize the 3D city model of Paris.

Integration of Geodata in Image-based 3D Geovisualization Applications

Typical applications using virtual 3D city models want to integrate different kinds of 2D and 3D geodata as well as to visualize georeferenced data to fulfill the application-specific and task-specific needs:

- Thematic maps can be used as terrain texture overlays (provided by Web Map Services);
- Planning variants retrieved as 3D models (provided by Web Feature Services);
- Georeferenced data originating from real-world sensors or the social web mapped on graphics variables of the virtual 3D city model.

Different strategies can be used to integrate that data into the visualization of virtual 3D city models. In general the data can be visualized by means of additional 3D model objects or by changing appearance or geometry of existing 3D model objects. There are three principal ways to implement these strategies (Fig. 5.):

- The data is mapped to 3D model elements added to the 3D scene by the 3D server system (server-based object-space mapping);
- The server maps and visualizes the data based on image post processing added to its rendering pipeline as a final step (server-based image-space mapping) [8];
- The data can be infiltrated directly into the client's 3D rendering process (client-based mapping).

The integration of application-specific data into the service-oriented visualization of virtual 3D city models can be managed by dedicated integration services. In particular, the image-based Web View Service is well suited as a core service to be used by such integration services because it supports server-based object-space and image-space mappings. The client-based mapping, however, requires modifications of the 3D client application and increases the costs for client-side 3D rendering.

The service-oriented visualization approach based on streaming of 3D panoramas, represents a promising strategy for future GIS and 3D geoinformation solutions. In particular, this approach encapsulates model storage, management, and rendering on the server side, relies on streaming fixed-size cube-maps, and enables interactive use of these virtual 3D panoramas on client applications. It can be extended by data integration services to adapt the visualization to the specific needs of an application. Most importantly, the approach allows us to take advantage of cloud computing and, therefore, is a fundamental building block for scalable 3D geovisualization systems.

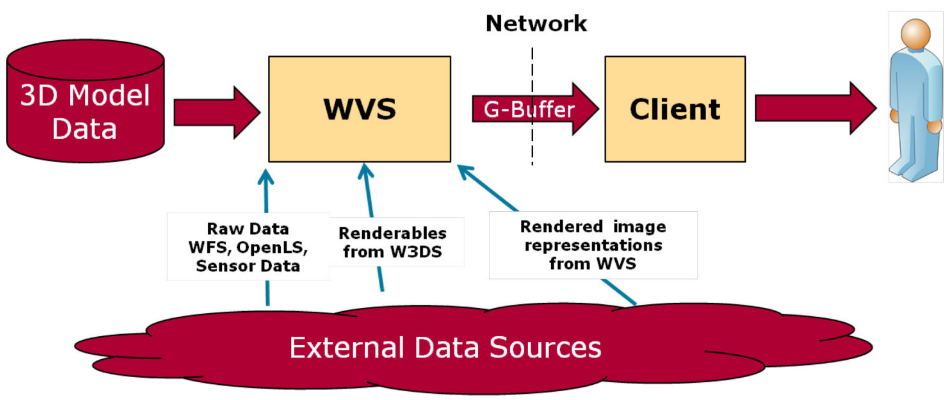

Figure 4 Types of data that can be used for integration into visualizations. Server side integration is shown, but similar techniquest can be implemented on client side.

References:

[1] Kolbe, Thomas H. "Representing and Exchanging 3D City Models with CityGML." In 3D Geo-Information Sciences, edited by Jiyeong Lee, Sisi Zlatanova, William Cartwright, Georg Gartner, Liqiu Meng, and Michael P Peterson, 15-31. Springer Berlin Heidelberg, 2009.[2] D. Hildebrandt, J. Klimke, B. Hagedorn, and J. Döllner, "Service-oriented interactive 3D visualization of massive 3D city models on thin clients" in *Proceedings of the 2nd International Conference on Computing for Geospatial Research & Applications - COM.Geo '11*, 2011, pp. 1-1.

[3] J. Engel and J. Döllner, "Approaches Towards Visual 3D Analysis for Digital Landscapes and Its Applications", Digital Landscape Architecture Proceedings 2009, 2009, pp. 33-41

[4] Homepage of the solar atlas Berlin, http://www.businesslocationcenter.de/en/3d-model-of-berlin/the-project/project-examples/solar-atlas

[5] B. Hagedorn, D. Hildebrandt, and J. Döllner, "Towards Advanced and Interactive Web Perspective View Services" in *Developments in 3D Geo-Information Sciences*, 2009, pp. 33-51.

[6] Web View Service Homepage, http://www.webviewservice.org/

[7] Web 3D Service Homepage, http://www.w3ds.org

[8] D. Hildebrandt, "Towards Service-Oriented, Standards- and Image-Based Styling of 3D Geovirtual Environments" in Proceedings of the 5th Ph.D. Retreat of the HPI Research School on Service-oriented Systems Engineering, 2011, pp. 133 - 147